Project 3: Sentiment Analysis

Divya Aggarwal (divyaagg830@gmail.com)

Exploratory Data Analysis

Data

The provided data is structured as a Pandas DataFrame with a RangeIndex of 1048572 entries, spanning from 0 to 1048571. The dataset consists of six columns, each serving a distinct purpose. The 'sentiment' column contains integer values, presumably representing sentiment labels such as positive or negative. The 'id' column also consists of integer values, serving as unique identifiers for each entry. The 'date' column is of object type, indicating that it likely contains date information. Similarly, the 'query' and 'user' columns are also of object type, suggesting they may store textual information related to queries and user identifiers, respectively. Finally, the 'text' column, also of object type, likely holds the textual content of the data, such as tweets or messages. It's noteworthy that all columns have non-null counts equal to the total number of entries, indicating a lack of missing values.

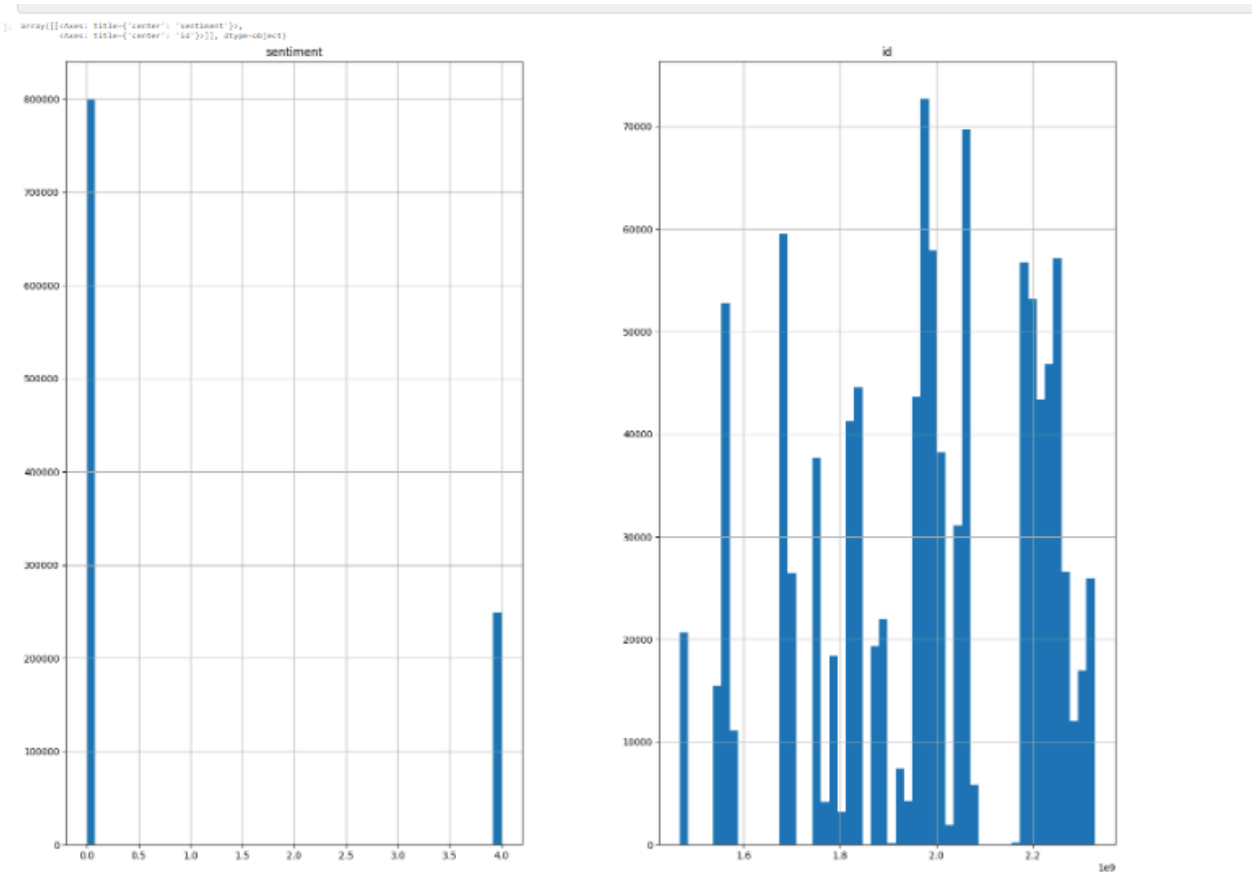
Data Description:

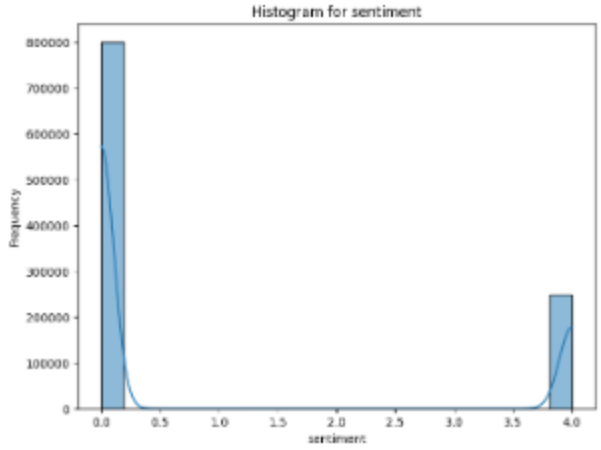
	sentiment	id
count	1.048572e+06	1.048572e+06
mean	9.482458e-01	1.976168e+09
std	1.701122e+00	2.300567e+08
min	0.000000e+00	1.467811e+09
25%	0.000000e+00	1.824526e+09
50%	0.000000e+00	1.990870e+09
75%	0.000000e+00	2.198903e+09
max	4.000000e+00	2.329206e+09

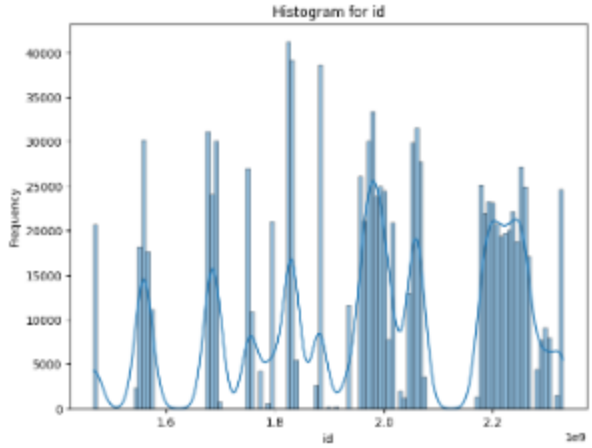
Unique Values

```
Column 'sentiment' has 2 unique values.
Column 'id' has 1048041 unique values.
Column 'date' has 662450 unique values.
Column 'query' has 1 unique values.
Column 'user' has 511364 unique values.
Column 'text' has 1036132 unique values.
```

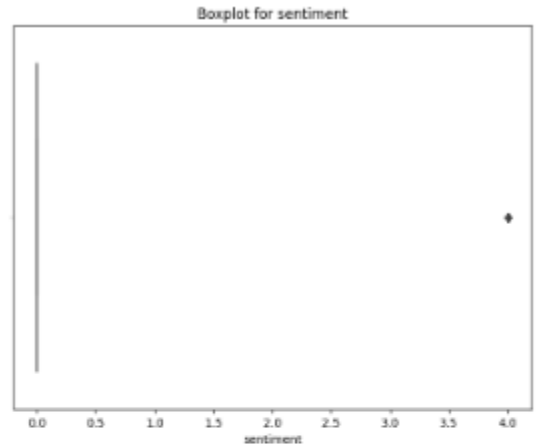
Histogram

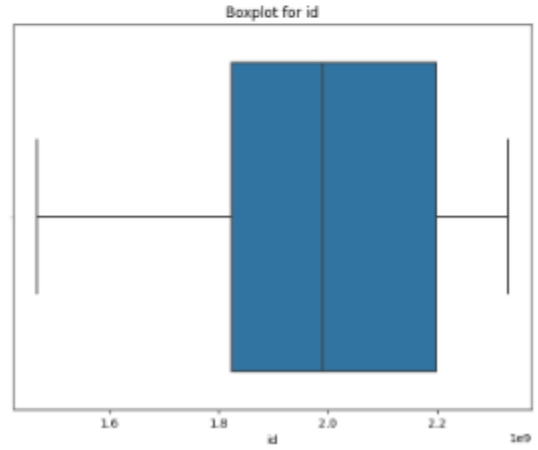



Histogram for sentiment
Frequency
sentiment


Histogram for id
Frequency
id
1e9

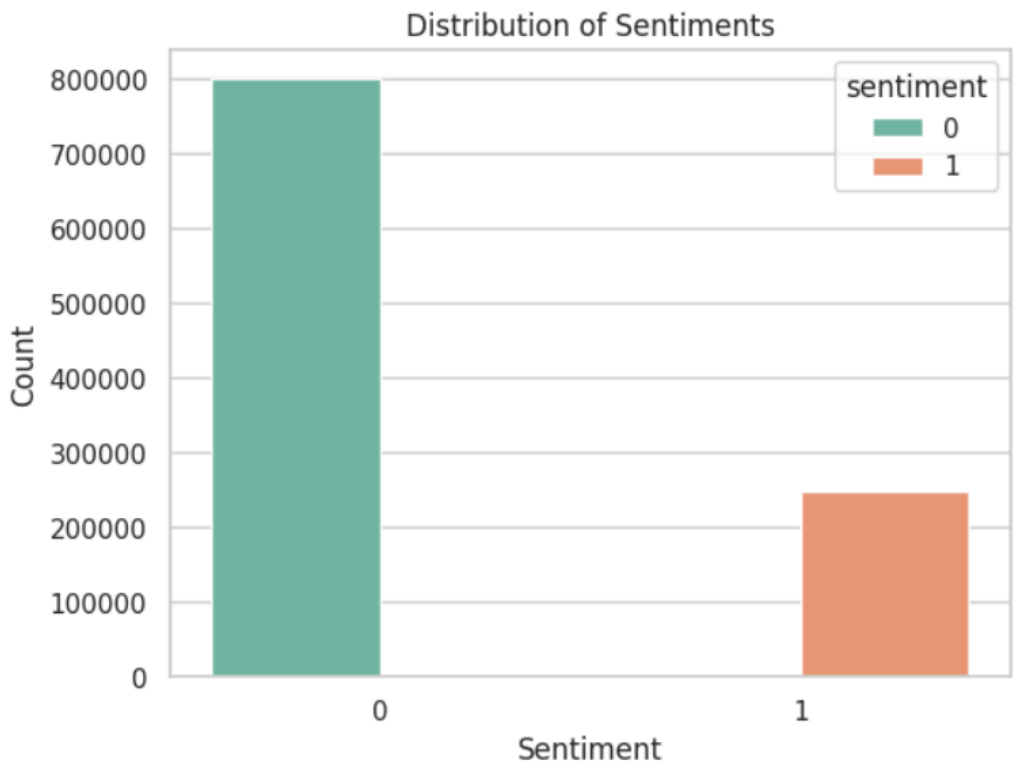
Boxplot and Outliers






```
Number of outliers in sentiment: 248576
Number of outliers in id: 0
```

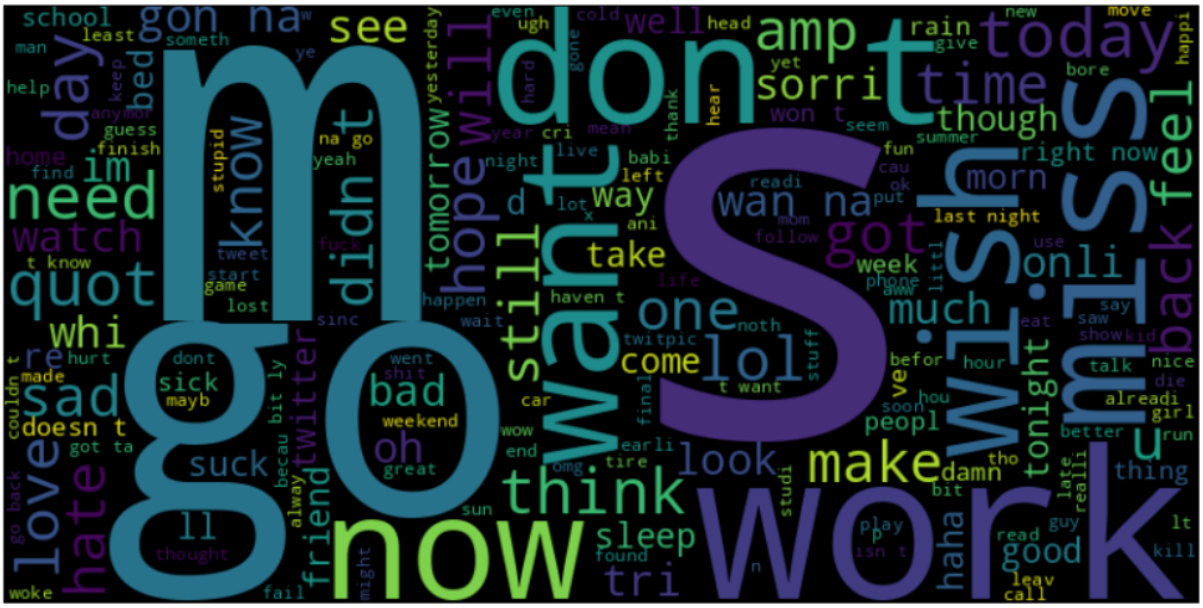
Distribution of Sentiments



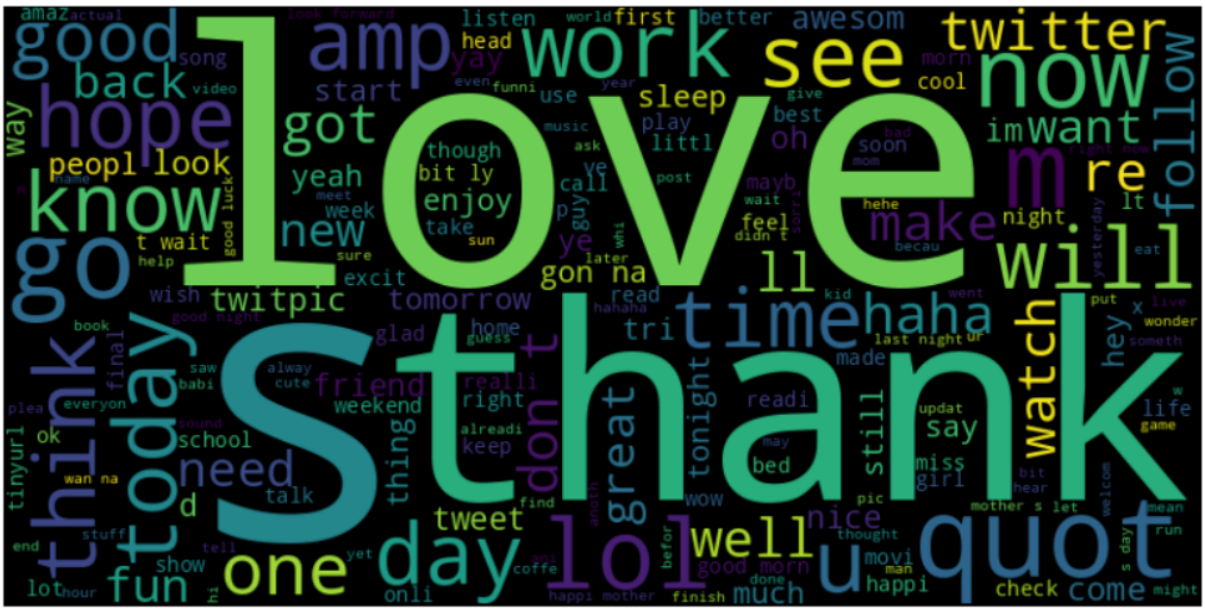
Preprocessing

Defined a text processing function named `textprocessing`, which is subsequently applied to the 'text' column of a DataFrame named 'df'. The function takes a text input, initially converting it to lowercase to ensure uniformity. It then employs regular expressions to remove hyperlinks, non-alphanumeric characters, and special symbols such as '@' and '#'. The function utilizes the NLTK library for tokenization, breaking down the text into individual words. Additionally, stemming is performed using the SnowballStemmer from NLTK to reduce words to their base or root form. During this process, common English language stop words and punctuation are removed. The final processed text is then joined into a single string. The resulting cleaned text is stored in a new DataFrame named 'data_cleaned' under the 'text' column. It's important to note that the specific NLTK library functions and regular expressions used in this text processing pipeline are tailored to clean and preprocess textual data, making it more amenable for natural language processing tasks or analysis.

Negative Text



Positive Text



Feature Scaling

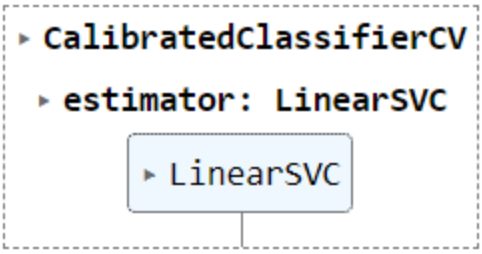
A sentiment analysis pipeline is being set up for machine learning. The text data from the 'text' column of the DataFrame 'df' is extracted and assigned to the variable x, while the corresponding sentiment labels are assigned to the variable y. Subsequently, a text vectorizer, TfidfVectorizer), is applied to convert the textual data into a matrix of TF-IDF features, stored in the variable vectors. The dataset is then split into a training set (70%) and a temporary set (30%) using the train_test_split function. The temporary set is further divided into validation and test sets, each comprising 50% of the temporary set. The choice of a random seed, specified as random_state=52, ensures reproducibility in the data split. Additionally, there is a comment indicating that the inclusion of n-grams may lead to a reduction in model accuracy, suggesting a thoughtful consideration of tokenization strategies and their impact on model performance.

SVM with PCA

Dimensionality reduction using Principal Component Analysis (PCA) is applied to the TF-IDF feature matrix obtained from the text data using Truncated Singular Value Decomposition (TruncatedSVD). The number of components is set to 100 (`n_components = 100`). The training, validation, and test sets are transformed into lower-dimensional representations using the fitted PCA model. A linear support vector machine (LinearSVC) model is then instantiated and trained on the reduced feature space of the training set (`X_train_svm_pca` and `y_train_svm`). Subsequently, predictions are made on both the validation and test sets, and accuracy scores are calculated by comparing the predicted labels with the true labels. The accuracy on the validation set is printed, providing an assessment of the model's performance on unseen data, and similarly, the accuracy on the test set is reported. This process integrates dimensionality reduction to enhance computational efficiency and focuses on evaluating the model's predictive accuracy on separate validation and test datasets.

```
Accuracy on the validation set: 0.7901042336893579
Accuracy on the test set: 0.7899819843006048
```

SVM without PCA



Train

```
Accuracy: 0.91
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.97      0.94    553132
           1       0.90      0.72      0.80    172160

    accuracy                           0.91    725292
   macro avg       0.91      0.84      0.87    725292
weighted avg       0.91      0.91      0.91    725292
```

The reported accuracy of 0.91 indicates the proportion of correctly predicted instances in the classification task. The classification report provides a more detailed evaluation of the model's performance, offering metrics such as precision, recall, and F1-score for each class. In this binary classification scenario, class 0 (presumably representing one sentiment class) demonstrates high precision (0.92), indicating a low false positive rate, and excellent recall (0.97), suggesting effective identification of true positives. Class 1, on the other hand, exhibits slightly lower precision (0.90) and recall (0.72), implying a trade-off between false positives and false negatives. The F1-score, which balances precision and recall, is notably high for class 0 (0.94) but lower for class 1 (0.80). The macro and weighted averages of precision, recall, and F1-score provide overall performance measures, with the macro average emphasizing class equality and the weighted average accounting for class imbalance. In summary, the model achieves a commendable accuracy of 0.91, with a detailed analysis revealing varying performance across the two sentiment classes.

Test

```
Accuracy: 0.85
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.94      0.90    118493
           1       0.75      0.53      0.62     36927

    accuracy                           0.85    155420
   macro avg       0.81      0.74      0.76    155420
weighted avg       0.84      0.85      0.84    155420
```

The reported accuracy of 0.85 indicates the proportion of correctly predicted instances in the classification task. The classification report provides a detailed breakdown of the model's performance for each class. For class 0, the precision is 0.87, indicating a low false positive rate, and the recall is 0.94, suggesting effective identification of true positives. The F1-score for class 0 is 0.90, representing a harmonic mean of precision and recall. Class 1 exhibits lower precision (0.75) and recall (0.53), indicating a trade-off between false positives and false negatives. The F1-score for class 1 is 0.62. The macro and weighted averages of precision, recall, and F1-score provide overall performance metrics, with the macro average emphasizing class equality and the weighted average accounting for class imbalance. In summary, the model achieves an accuracy of 0.85, with a more detailed analysis revealing varying performance across the two classes, particularly with lower precision and recall for class 1.

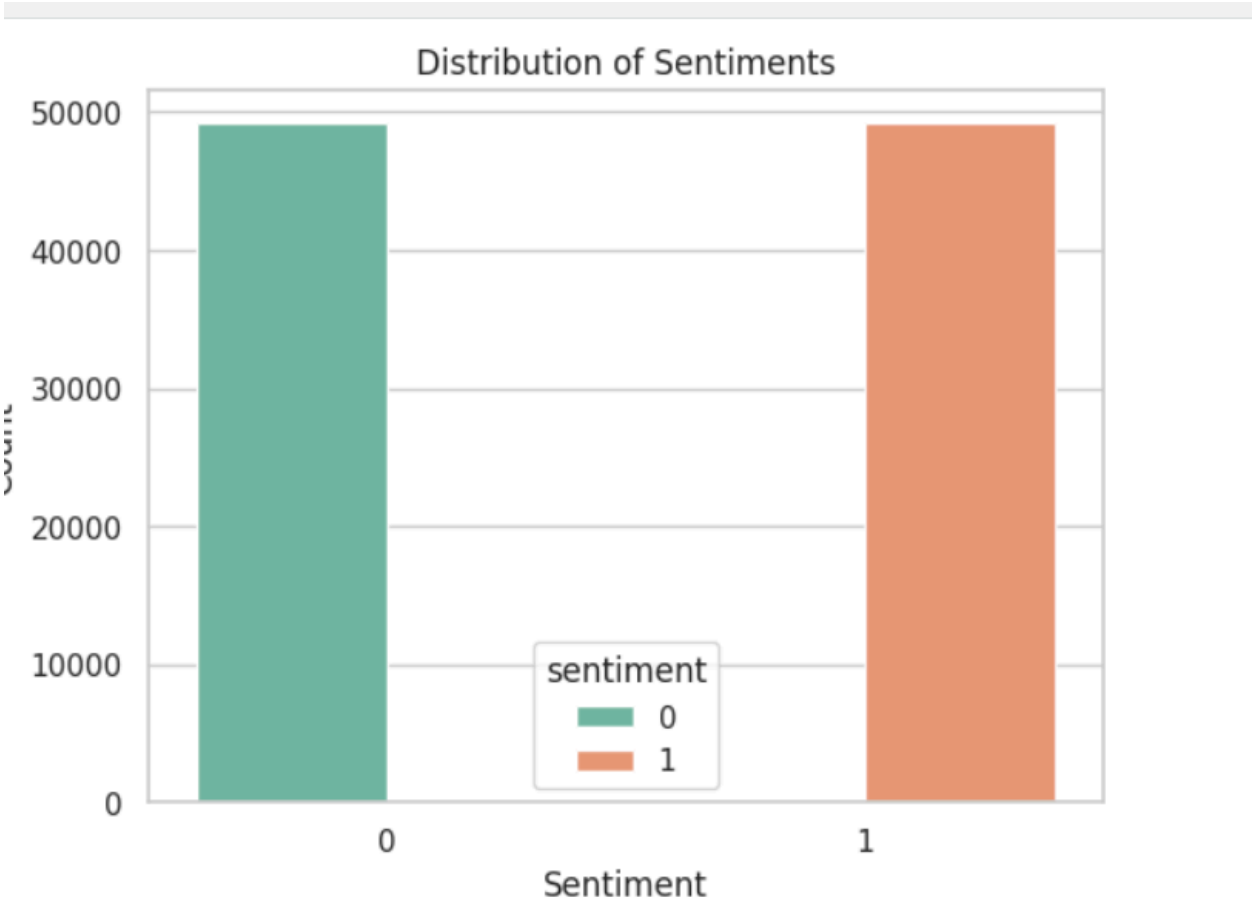
Validation

```
Accuracy: 0.85
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.95      0.90    118556
           1       0.75      0.53      0.62     36864

    accuracy                           0.85    155420
   macro avg       0.81      0.74      0.76    155420
weighted avg       0.84      0.85      0.84    155420
```

The reported accuracy of 0.85 indicates the proportion of correctly predicted instances in the classification task. The classification report provides a detailed breakdown of the model's performance for each class. For class 0, the precision is 0.87, indicating a low false positive rate, and the recall is 0.95, suggesting effective identification of true positives. The F1-score for class 0 is 0.90, representing a harmonic mean of precision and recall. Class 1 exhibits lower precision (0.75) and recall (0.53), indicating a trade-off between false positives and false negatives. The F1-score for class 1 is 0.62. The macro and weighted averages of precision, recall, and F1-score provide overall performance metrics, with the macro average emphasizing class equality and the weighted average accounting for class imbalance. In summary, the model achieves an accuracy of 0.85, with a more detailed analysis revealing varying performance across the two classes, particularly with lower precision and recall for class 1. The reported metrics appear consistent with the previous classification report, suggesting a stable performance of the model across different evaluations.

Downsampling

An approach to address class imbalance in the sentiment analysis dataset is implemented using downsampling. The majority class (class 0) and minority class (class 1) are separated into two dataframes, namely `df_majority_knn` and `df_minority_knn`, respectively. Subsequently, downsampling is performed on both classes using the `resample` function from the scikit-learn library. The downsampling is set to replace=False, ensuring that the sampled instances are not replaced in the original dataframes. The number of samples for both majority and minority classes is controlled by the parameter `n_samples`, set here to one-fifth of the original size of the minority class. The random seed is specified as `random_state=134` for reproducibility. Finally, the downsampled dataframes for both majority and minority classes are concatenated using `pd.concat`, resulting in a more balanced dataset with reduced instances of the majority class. The text data is then vectorized using a text vectorizer (`vectorizer2`) to prepare it for machine learning model training. The shape of the resulting TF-IDF feature matrix is obtained and can be used as input for subsequent model training to potentially mitigate the impact of class imbalance on the model's performance.



KNN with PCA

Dimensionality reduction using Principal Component Analysis (PCA) is applied to the TF-IDF feature matrix obtained from the downsampled text data. The number of components is set to 100 (`n_components = 100`). The training, validation, and test sets are transformed into lower-dimensional representations using the fitted PCA model (`svd`). Subsequently, a k-nearest neighbors (KNN) classification model is instantiated with `n_neighbors=5` and trained on the reduced feature space of the training set (`X_train_knn_pca` and `y_train_knn`). Predictions are then made on both the validation and test sets, and accuracy scores are calculated by comparing the predicted labels with the true labels. The accuracy on the validation set is printed, providing an assessment of the model's performance on unseen data, and similarly, the accuracy on the test set is reported. This process integrates both dimensionality reduction and downsampling techniques to potentially address class imbalance in the dataset and evaluate the performance of the KNN classifier in this modified setting.

```
Accuracy on the validation set: 0.6607033949989836
Accuracy on the test set: 0.6463373314359288
```

KNN without PCA

Train

```
KNN Accuracy: 0.74
KNN Classification Report:
              precision    recall  f1-score   support

           0       0.70      0.85      0.77     34502
           1       0.81      0.63      0.71     34364

    accuracy                           0.74     68866
   macro avg       0.75      0.74      0.74     68866
weighted avg       0.75      0.74      0.74     68866
```

The reported accuracy of 0.74 for the K-nearest neighbors (KNN) classifier indicates the proportion of correctly predicted instances in the binary sentiment analysis task. The accompanying classification report provides a detailed breakdown of the model's performance for each class. For class 0, the precision is 0.70, indicating a low false positive rate, and the recall is 0.85, suggesting effective identification of true positives. The F1-score for class 0 is 0.77, representing a harmonic mean of precision and recall. Class 1 exhibits higher precision (0.81) and lower recall (0.63), indicating a trade-off between false positives and false negatives. The F1-score for class 1 is 0.71. The macro and weighted averages of precision, recall, and F1-score provide overall performance metrics, with the macro average emphasizing class equality and the weighted average accounting for class imbalance. In summary, the KNN classifier achieves an accuracy of 0.74, with varying performance across the two sentiment classes, particularly with higher recall for class 0 and higher precision for class 1. The reported metrics offer insights into the model's strengths and limitations in handling the downsampled and dimensionality-reduced dataset.

Test

```
KNN Accuracy: 0.74
KNN Classification Report:
              precision    recall  f1-score   support

           0       0.70      0.85      0.76      7324
           1       0.81      0.63      0.71      7433

    accuracy                           0.74     14757
   macro avg       0.75      0.74      0.74     14757
weighted avg       0.75      0.74      0.74     14757
```

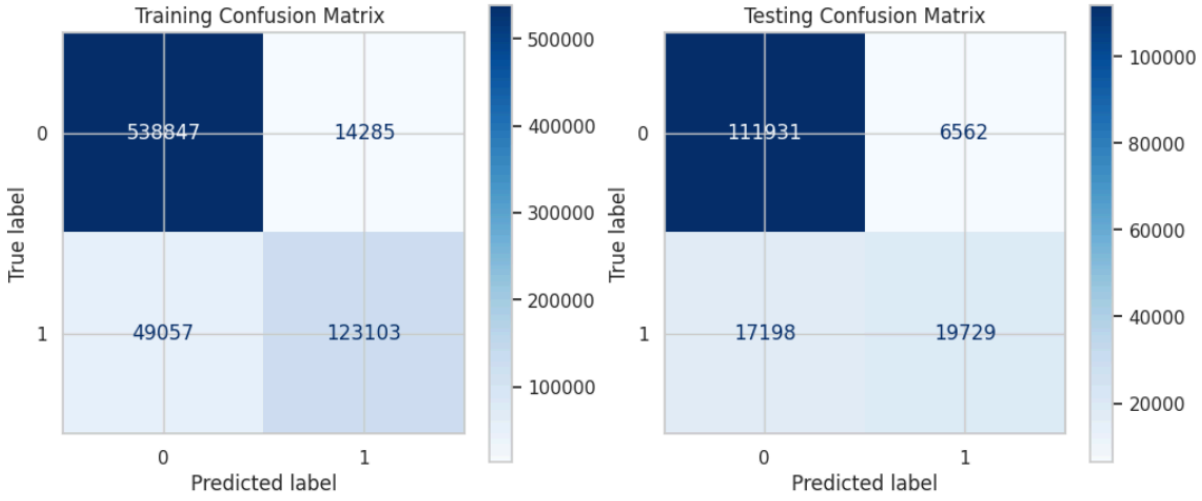
The reported accuracy of 0.74 for the K-nearest neighbors (KNN) classifier indicates the proportion of correctly predicted instances in the binary sentiment analysis task. The accompanying classification report provides a detailed breakdown of the model's performance for each class on the validation set. For class 0, the precision is 0.70, indicating a low false positive rate, and the recall is 0.85, suggesting effective identification of true positives. The F1-score for class 0 is 0.76, representing a harmonic mean of precision and recall. Class 1 exhibits higher precision (0.81) and lower recall (0.63), indicating a trade-off between false positives and false negatives. The F1-score for class 1 is 0.71. The macro and weighted averages of precision, recall, and F1-score provide overall performance metrics, with the macro average emphasizing class equality and the weighted average accounting for class imbalance. In summary, the KNN classifier achieves an accuracy of 0.74 on the validation set, with varying performance across the two sentiment classes, particularly with higher recall for class 0 and higher precision for class 1. The reported metrics offer insights into the model's strengths and limitations in handling the downsampling and dimensionality-reduced dataset on the validation set.

Validation

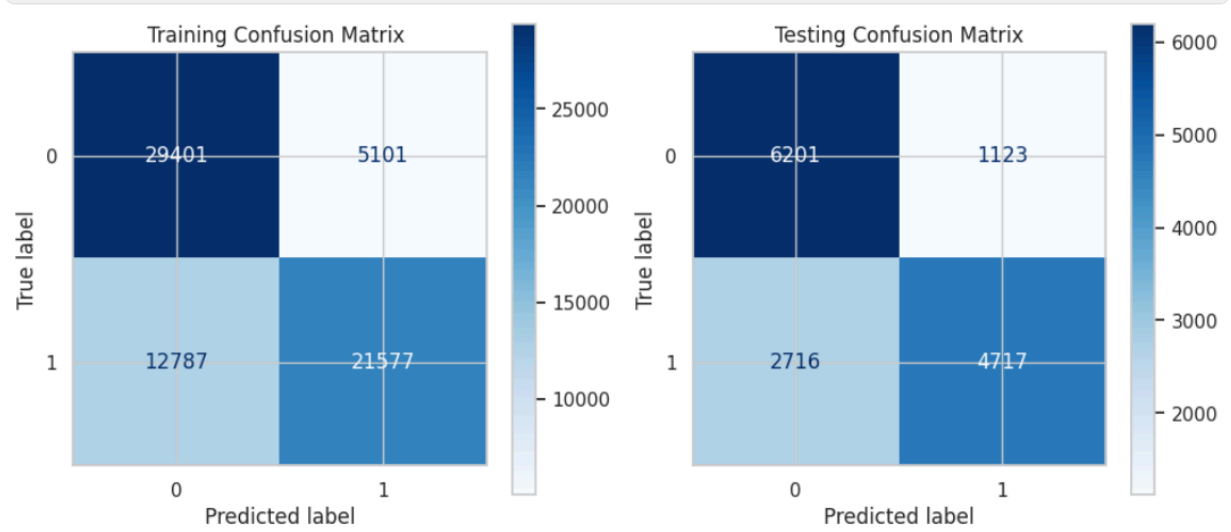
```
KNN Accuracy: 0.73
KNN Classification Report:
              precision    recall  f1-score   support

           0       0.69      0.85      0.76      7364
           1       0.81      0.62      0.70      7393

    accuracy                           0.73     14757
   macro avg       0.75      0.74      0.73     14757
weighted avg       0.75      0.73      0.73     14757
```

The reported accuracy of 0.73 for the K-nearest neighbors (KNN) classifier indicates the proportion of correctly predicted instances in the binary sentiment analysis task on the validation set. The accompanying classification report provides a detailed breakdown of the model's performance for each class. For class 0, the precision is 0.69, indicating a relatively low false positive rate, and the recall is 0.85, suggesting effective identification of true positives. The F1-score for class 0 is 0.76, representing a harmonic mean of precision and recall. Class 1 exhibits higher precision (0.81) and lower recall (0.62), indicating a trade-off between false positives and false negatives. The F1-score for class 1 is 0.70. The macro and weighted averages of precision, recall, and F1-score provide overall performance metrics, with the macro average emphasizing class equality and the weighted average accounting for class imbalance. In summary, the KNN classifier achieves an accuracy of 0.73 on the validation set, with varying performance across the two sentiment classes, particularly with higher recall for class 0 and higher precision for class 1. The reported metrics offer insights into the model's strengths and limitations in handling the downsampled and dimensionality-reduced dataset on the validation set.

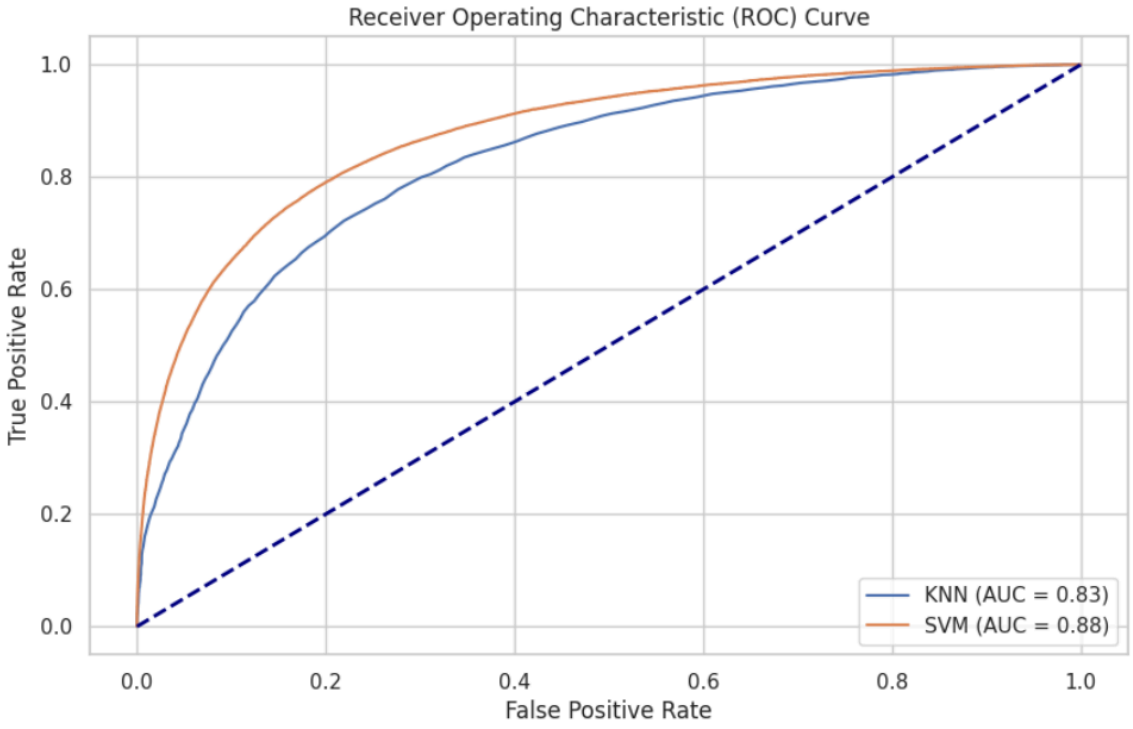
SVM Confusion Matrix



KNN Confusion Matrix



ROC Curve



Conclusion

Model	Training Accuracy	Testing Accuracy	Validation Accuracy
SVM	91%	85%	85%
KNN	75%	74%	74%

The SVM results showed 91% training accuracy, 85% testing accuracy and 85% validation accuracy. The KNN shows 75% training accuracy , 74% testing accuracy and 74% validation accuracy.